

需要检视的核心代码比较简单，主要是用代理模式对软硬件编解码器以及混屏器做封装，目的是屏蔽差异，大部分代码主要是实现一个代理调用，少数要做区别判断。

质量属性分类		序号	描述		得分
软件质量 (外部质量)	功能	1	一般	MediaFactory.UnInitFactory中，日志最后释放较好点，否则释放期间的日志打印不出来。	20
		2	一般	MediaFactory.DestroyXX代码，如果二级指针为空，会导致一级指针不会被释放。	
		3	一般	MediaFactory::CreateSource，未判空	
		4	一般	MediaRender::CreateRender，日志输出代码缺参数。	
		5	一般	MediaFactory.CreateXX代码，函数返回失败时，不应该携带有效出参，更不应该让外部确保释放资源。	
	性能	1		代码时间性能：使用多个锁减少冲突的产生；多线程做生产消费协作，利用条件变量阻塞等待。时间性能上无明显短板	10
		2		空间性能：主要的消耗在生产消费中的缓冲队列，其中使用循环队列，有效的兼顾了去抖动以及对时延的要求，值得借	
	其它				
代码质量 (内部质量)	代码设计	1		(容错性)	
		2		软硬解码器输出物有明显的差异性，对使用者也都有不一样的要求，而IDecoder接口设计并没有起到屏蔽差异的作用（同样的问题也存在于编码器封装）。使用接口统一封装有点勉强，适用性比较弱，建议从更高层次封装（比如把解码器和编码器一并封装）（里氏替换，依赖倒置）	
		3			
	代码实现			符合，无明显问题	10
					19
	代码坏味道			std::unique_lock<std::mutex> lock(m_decoder_mtx);这种代码比较多，稍显不那么优雅，可以用宏定义下。（重复的代码）	